Social Networks

Università degli Studi di Salerno

Fisciano
July 22, 2016

Speakers:

Rosario Di Florio, Emanuele Russomanno, Vincenzo Venosi

Introduction

The purpose of our work is to test different algorithms in the three fundamental areas for assembling any search engine offering a Sponsored Search system:

- **Ranking** of web documents
- **Matching** of words inside documents
- **Auctions** for acquiring advertisement slots

We will talk about the proposed algorithms, and then compare running times and results obtained from their execution more in detail, suggesting what combination of algorithms seems to be the best for realizing a new search engine.

Overview

1. Creating Dataset
2. Ranking
   a. Page Rank
   b. HITS
   c. Comparing the Results
3. Matching
   a. Best Match
   b. Optimization of Best Match
   c. Results
4. Search Engine
   a. Results
5. Auctions
   a. First Price Auction
   b. Generalized Second Price Auction
   c. Results
6. Conclusion

Creating the Dataset

- Our experiments ran on a set of approximatively 30000 pages created in this way: We have choose a web-page for each of the 15 categories listed in: https://www.dmoz.org/
- For every of these web pages we crawled 2000 pages by using the Wibbi online crawler http://wb28.stanford.edu/~testbed/cgi-bin/crawlStreamingControls.pl
- From each pair of sets of 2000 pages, we choose 10 random pairs of vertices (u, v), with u being a page in the first set and v being a page in the second set and added a link from u to v (if this link was absent)

Creating the Dataset

The links used are shown below:

Category	Website	Description
Arts	www.awn.com/	Provides information resources to the international animation community.
Business	www.irs.gov/	The IRS is the U.S. government agency responsible for tax collection and tax law enforcement. Contains downloadable forms, instructions, and agency publications. It also includes "The Digital Daily," an almost-humorous online newspaper.
Computers	www.tomshardware.com/	All kinds of technical product reviews including motherboards, CPUs, memory chips and video cards.
Games	www.ign.com/	Games news, previews and behind the scenes information
Health	www.webmd.com/	Resource for consumers, physicians, nurses, and educators. Includes news, chat forums, health quizzes and consumer product updates.
Home	www.groupon.com/	Features a daily deal for most major metropolitan areas in the United States.
Kid and Teens	www.seventeen.com/	Talks about hair, skin, and make-up, dating, health, college, and career.
News	abcnews.go.com/	Includes American and world news headlines, articles, chatrooms, message boards, news alerts, video and audio webcasts, shopping, and wireless news service. As well as news television show information and content.

Creating the Dataset

The links used are shown below:

Category	Website	Description
News	abcnews.go.com/	Includes American and world news headlines, articles, chatrooms, message boards, news alerts, video and audio webcasts, shopping, and wireless news service.
Recreation	www.autoblog.com/	Weblog grasping the auto industry with test drives and commentary on articles from other sites
Reference	stackoverflow.com/	A language-independent collaboratively edited question and answer site for programmers. Questions and answers displayed by user votes and tags.
Regional	www.governo.it/	Official site of italian government. Here are showed the government activities.
Science	www.nature.com/	Nature.com provides access to all Nature Publishing Group journals, online databases and services, including Nature News, and the social network for scientists Nature Network.
Shopping	www.amazon.com/	Amazon.com seeks to be Earth's most customer-centric company, where customers can find and discover anything they might want to buy online, and endeavors to offer its customers the lowest possible prices.
Society	www.ancestry.com/	Talks about hair, skin, and make-up, dating, health, college, and career.
Sport	espn.go.com/	Sports news network. Includes broadcast schedule, game scores and results, and articles on college and professional sports.

Ranking

- Page Rank
  - Results
- HITS
  - Results
- Comparing Algorithms

PageRank

The intuition behind Page Rank is:

“a page is important if it is cited by other important pages”.

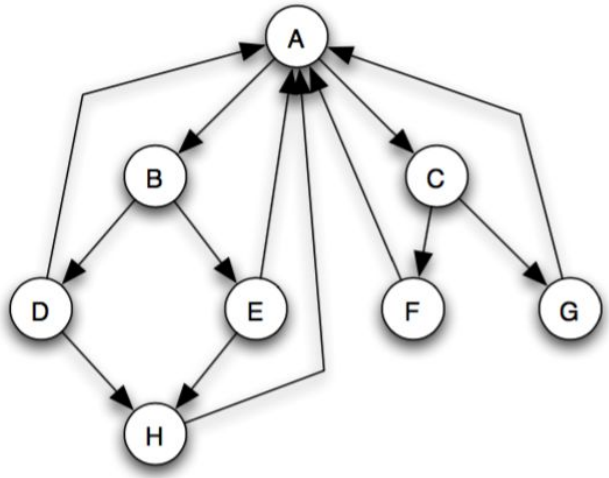
Basically it’s start with simple voting based on in-going links, and refines it using the Priciples of Repeated Improvement, that is based on the repeated passage of endorsements across node’s out-going links.

PageRank

We can think of PageRank as a kind of “fluid” that circulates through the network, passing from node to node across edges, and pooling at most important nodes.

PageRank is computed as follow:

- In a network with n nodes, we assign all nodes the same initial PageRank,set to be 1/n.
- We choose a number of steps k.
- We then perform a sequence of k updates to the PageRank values.



PageRank

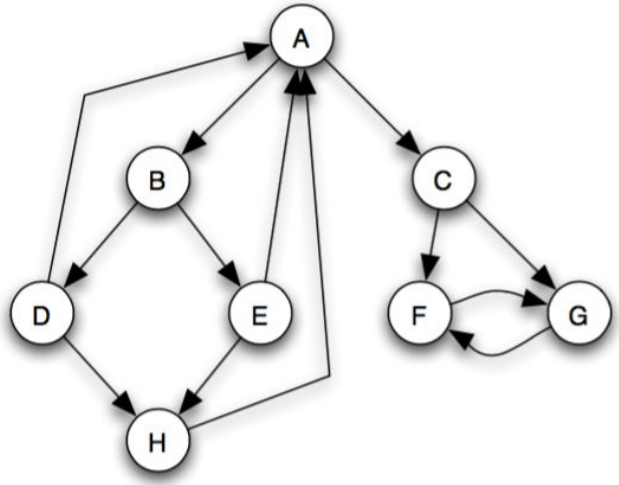
Basic PageRank Update Rule:

Each page divides its current PageRank equally across its out-going links, and passes these equal shares to the pages it points to. Each page updates its new PageRank to be the sum of the shares it receives.

There is a difficulty with the basic definition of PageRank, however: in many networks, the “wrong” nodes can end up with all the PageRank.

PageRank

The Wrong nodes are a small sets of nodes that can be reached from the rest of the graph, but have no paths back.



Scaled Pagerank Update Rule:

First apply the Basic PageRank Update Rule. Then scale down all PageRank values by a factor of s. This means that the total PageRank in the network has shrunk from 1 to s. We divide the residual 1 − s units of PageRank equally over all nodes, giving (1 − s)/n to each.

PageRank

Experiment configurations

We are going to present the result of the experiment, comparing the execution time of PageRank and average number of covergence steps on the following inputs (generated by chunking the Full Graph) and different confidences:

Tested graph's size (with incremental step of 1000 nodes):

- 1000
- 2000
- ….
- ….
- 29000
- 29995

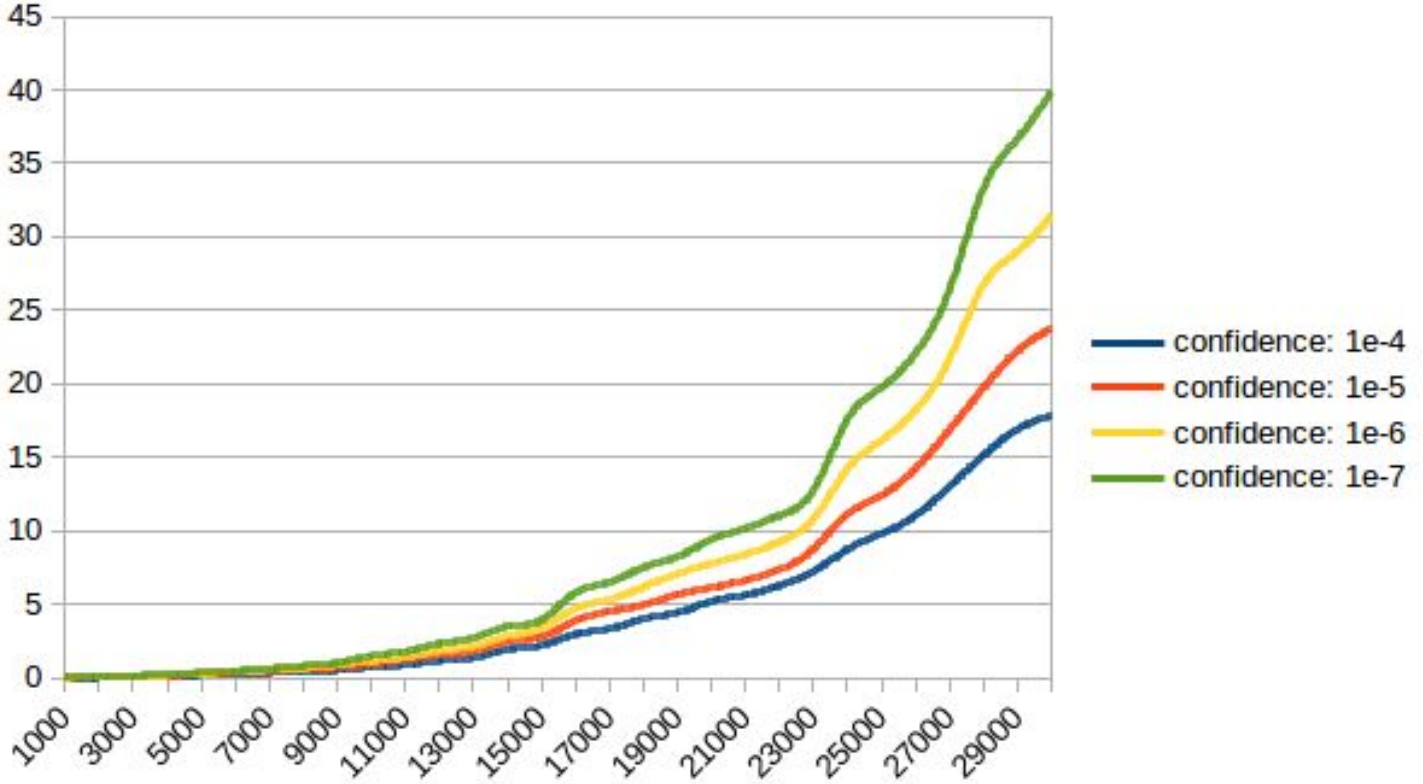
Tested confidences :

- $1e^{-4}$
- $1e^{-5}$
- $1e^{-6}$
- $1e^{-7}$

each test was iterated for 20 times

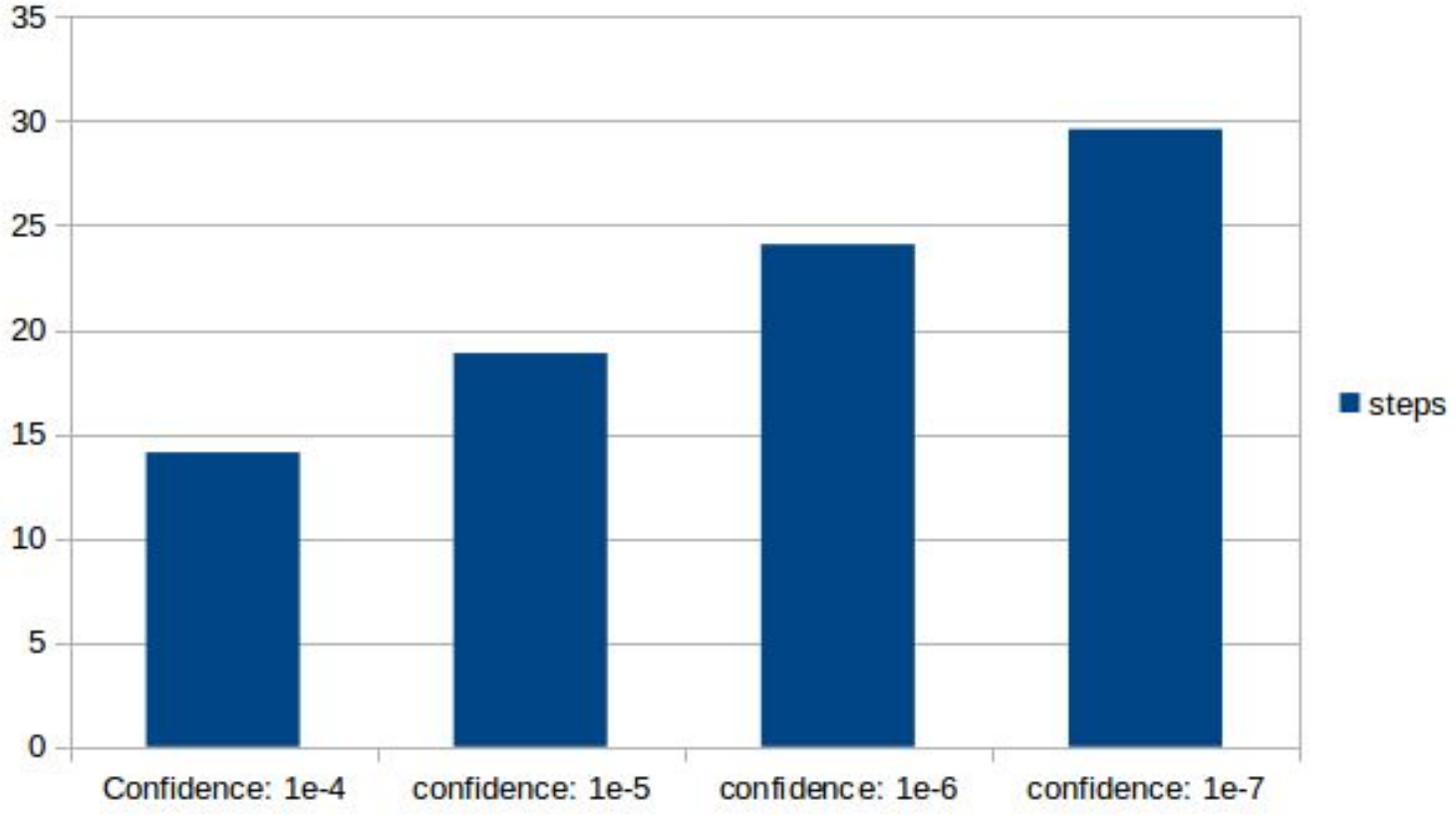
PageRank

Results - Times



PageRank

Results - Average number of steps



HITS

An new idea based “**hubs and authorities**” was proposed shortly after PageRank was first implemented.

This hubs-and-authorities algorithm, sometimes called **HITS** (hyperlinkinduced topic search), was originally intended not as a preprocessing step before handling search queries, as PageRank is, but as a step to be done along with the processing of a search query, to rank only the responses to that query.

HITS

The intuition behind HITS is:

**“a page is a good hub if it links to good authorities,
and a page is a good authority if it is linked to by good hubs”.**

While PageRank assumes a one-dimensional notion of importance for pages, HITS views important pages as having two flavors of importance:

1. Certain pages are valuable because they provide information about a topic. These pages are called **authorities**.
2. Other pages are valuable not because they provide information about any topic, but because they tell you where to go to find out about that topic. These pages are called **hubs**.

HITS - Algorithm

The **algorithm** assign two scores to each Web page. One score represents the hubbiness of a page, that is the degree to which it is a good hub, and the second score represents the degree to which the page is a good authority.

These values are then calculated as follow:

Hubbiness: the sum of the Authority value of the outgoing nodes.

Authority: the sum of Hubbiness value of the incoming nodes.

Typically these values would grow beyond bounds, so they are scaled so that the largest value is 1.

HITS - Improvement

On the first attempts of running the algorithm on the full graph we observed that one iteration took about 30 minutes, due to the nature of the algorithm. In each iteration we explore all the graph and calculate the incoming nodes for the current node...

Considering that the graph never changes, we precomputed all the incoming nodes for each node so we can obtain the incoming nodes in O(1).

HITS

Experiment configurations

We are going to present the result of the experiment, comparing the execution time and average number of covergence steps of HITS on the following inputs and different confidences:

Tested graph's size (with incremental step of 1000 nodes):

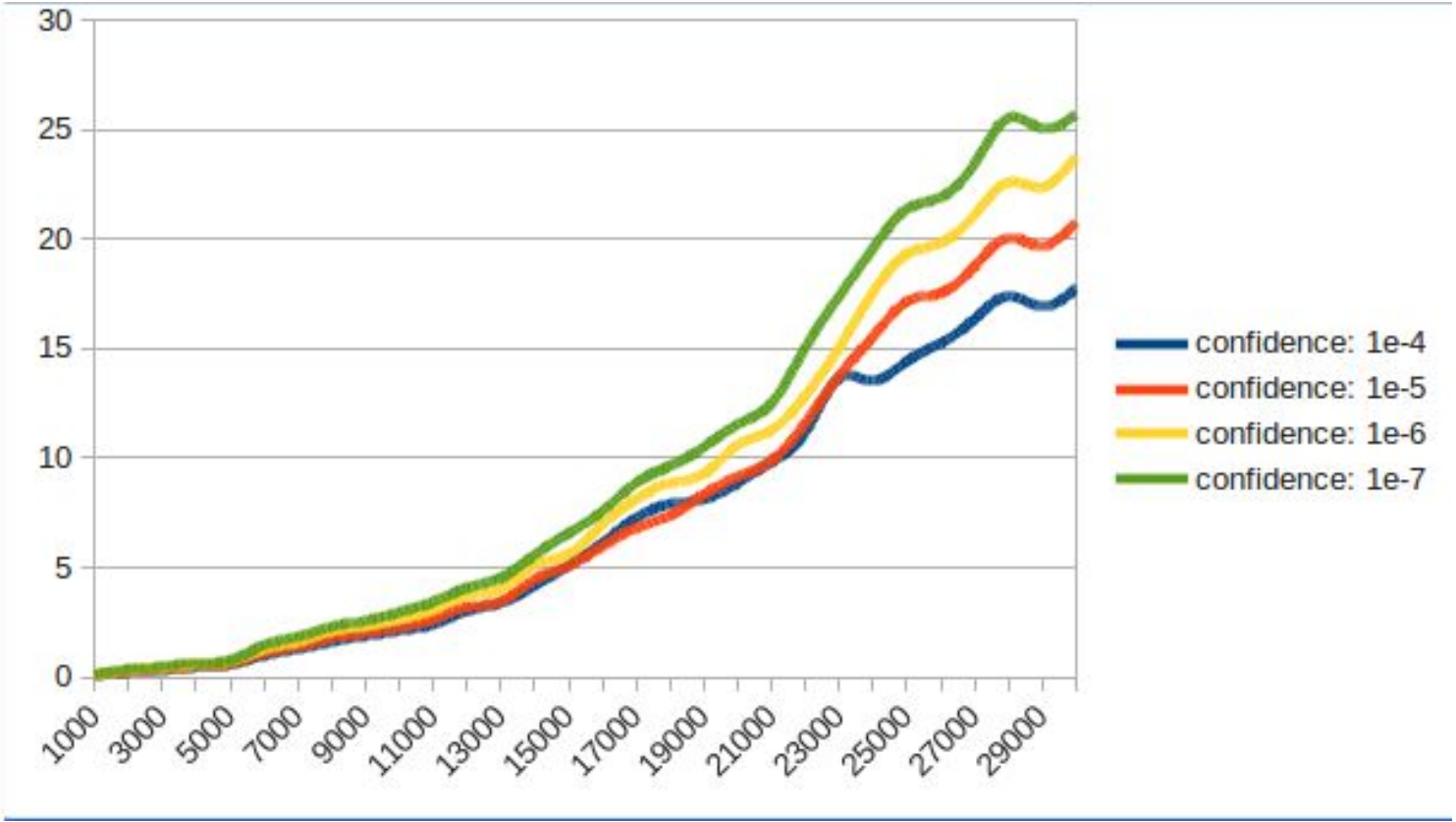
- 1000
- 2000
- ….
- ….
- 29000
- 29995

Tested confidences :

- $1e^{-4}$
- $1e^{-5}$
- $1e^{-6}$
- $1e^{-7}$

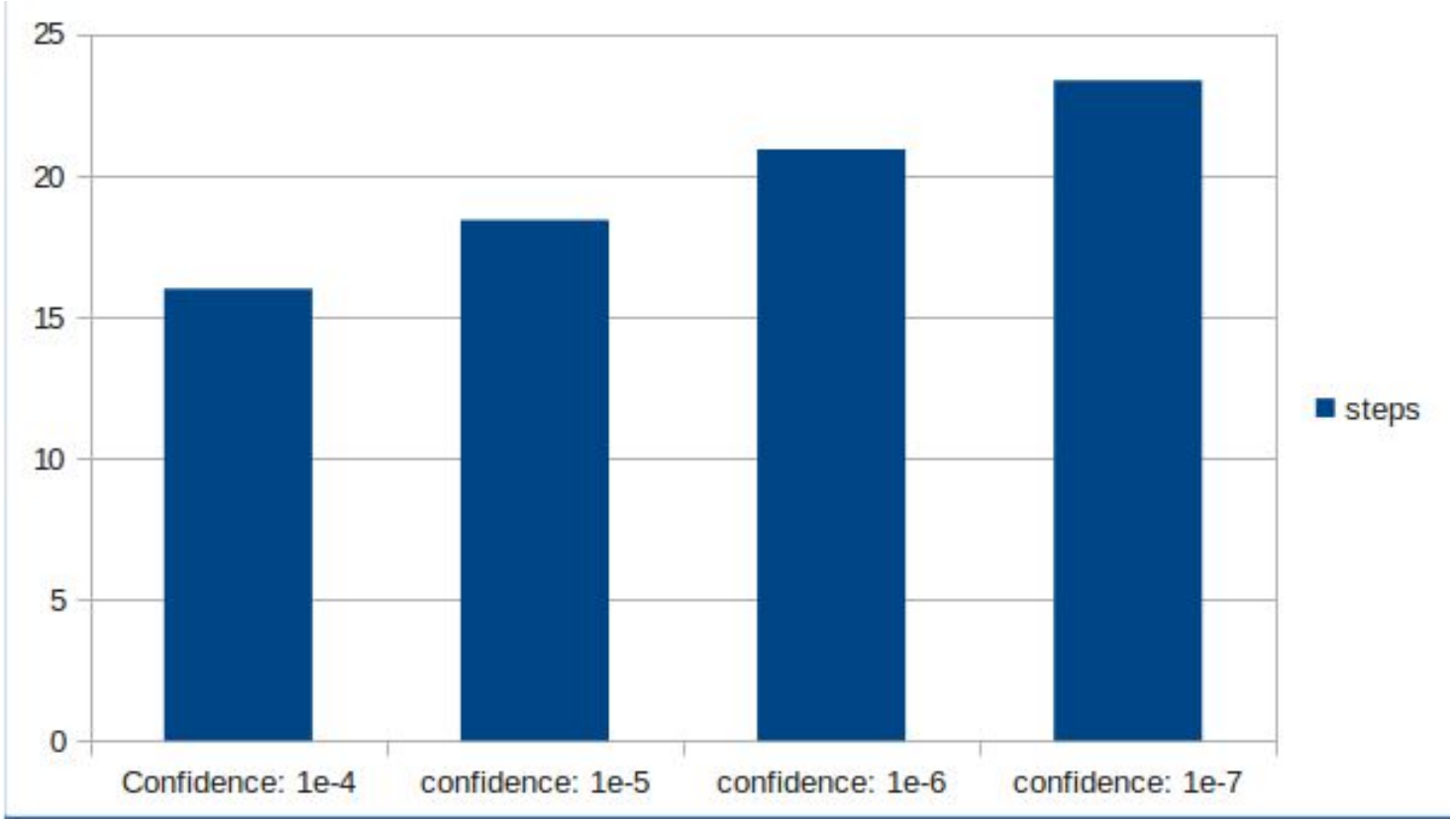
HITS

Results - Times



HITS

Results - Average number of steps



PageRank

V

S

HITS

PageRank vs HITS

Rank values

The rank values of pagerank and HITS normally are not comparable because these values are on different scales.

We have created an heuristic to see the importance that the two algorithms give the documents of our dataset.

- We have sorted results of the different algoritms.
  - es. index 0 for PageRank results is the most important page for the algorithm(the same for HITS with the different values).
  - to each index $\mathbf{page_i}$ starting from 0 we assign it **a - i/b** where:
    - **a = b = |pages|**

PageRank vs HITS

Rank values

For the nature of these algorithms, the results giving different information on the pages.
So the importance of the pages changes for each algorithm.

However we have seen that HITS autority and PageRank have the same behavior in a lot of cases:

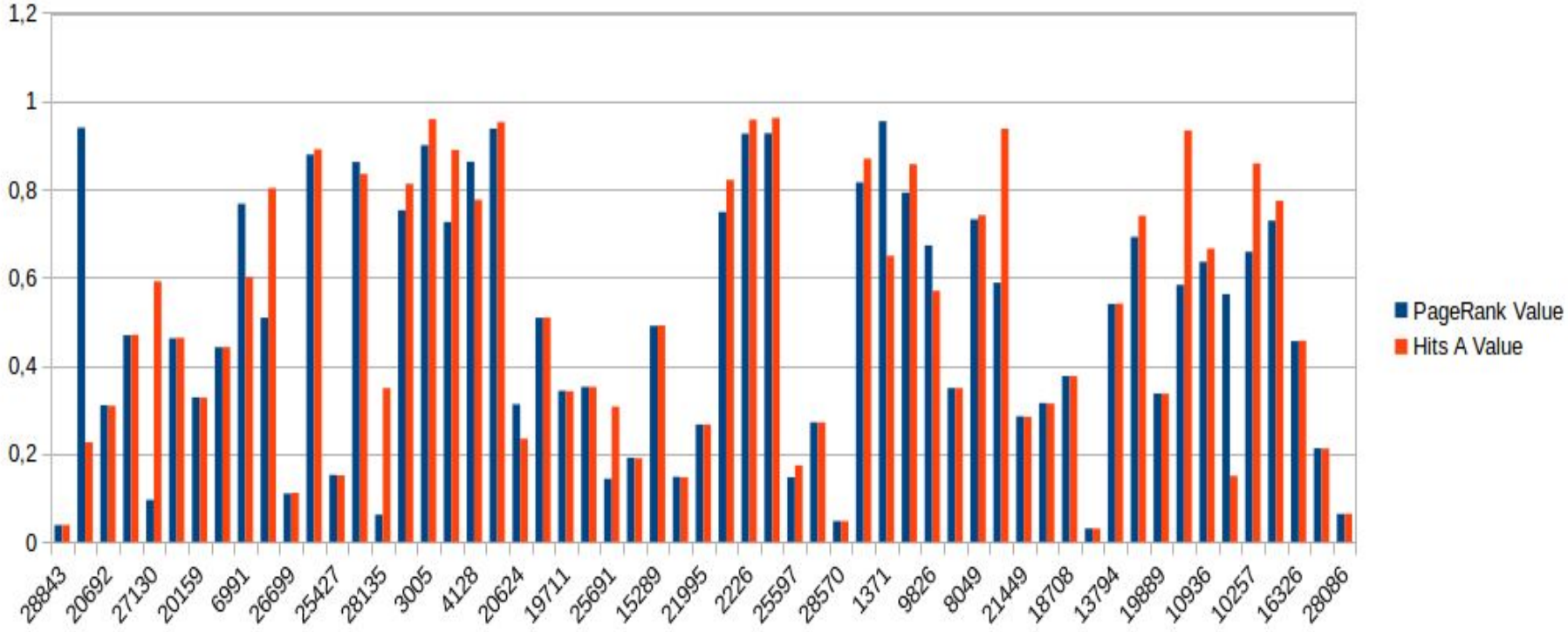
PageRank:

"a page is important if it is cited by other important pages"

HITS Autority:

"a page is a good authority if it is linked to by good hubs "

PageRank vs HITS



PageRank vs HITS

HITS weighted

HITS give more informations about the page with the hubbiness.

This value is a helpful information because points out the pages where we can find authoritative documents about the topic more easily , but we can't use only this value to a search engine.

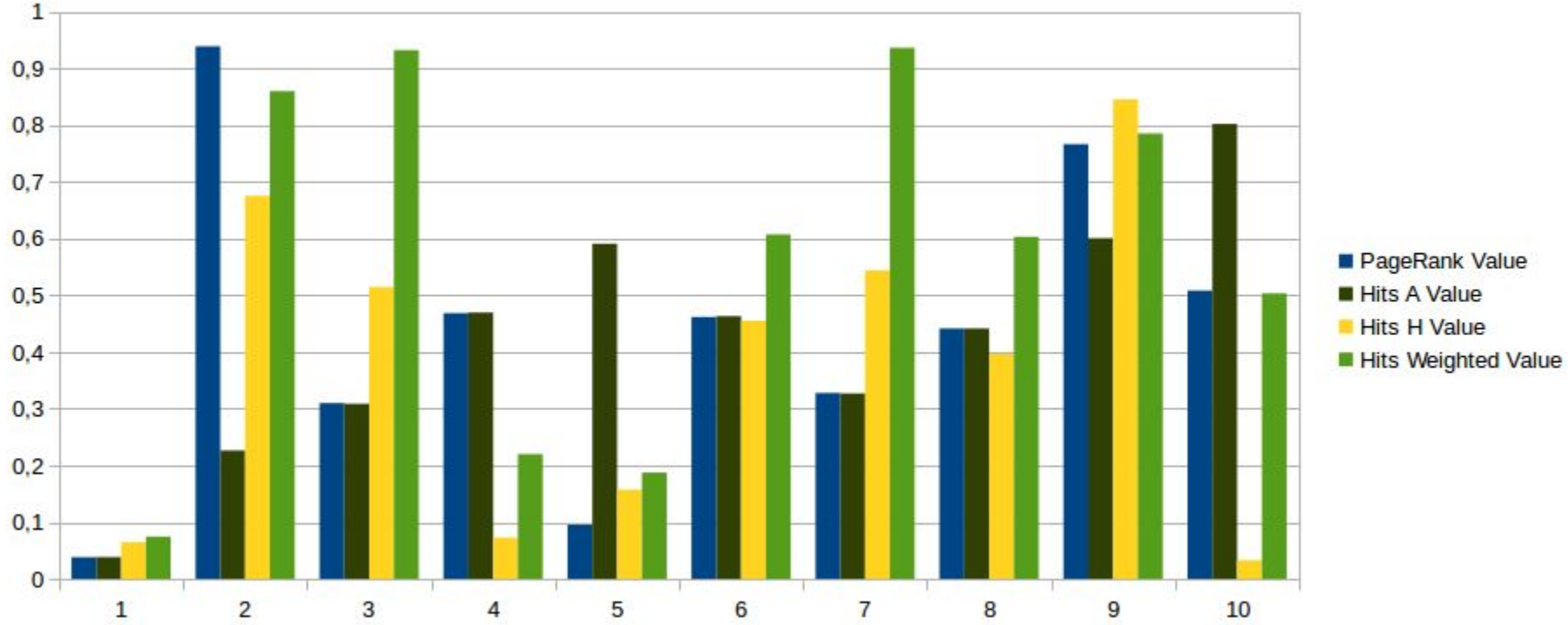
So we have created a weighted sum between HITS authority and HITS hubbiness

HITS w = (authority * 0.6) + (hubbiness* 0.4)

Following we show a little pool of nodes where we see the different node rank

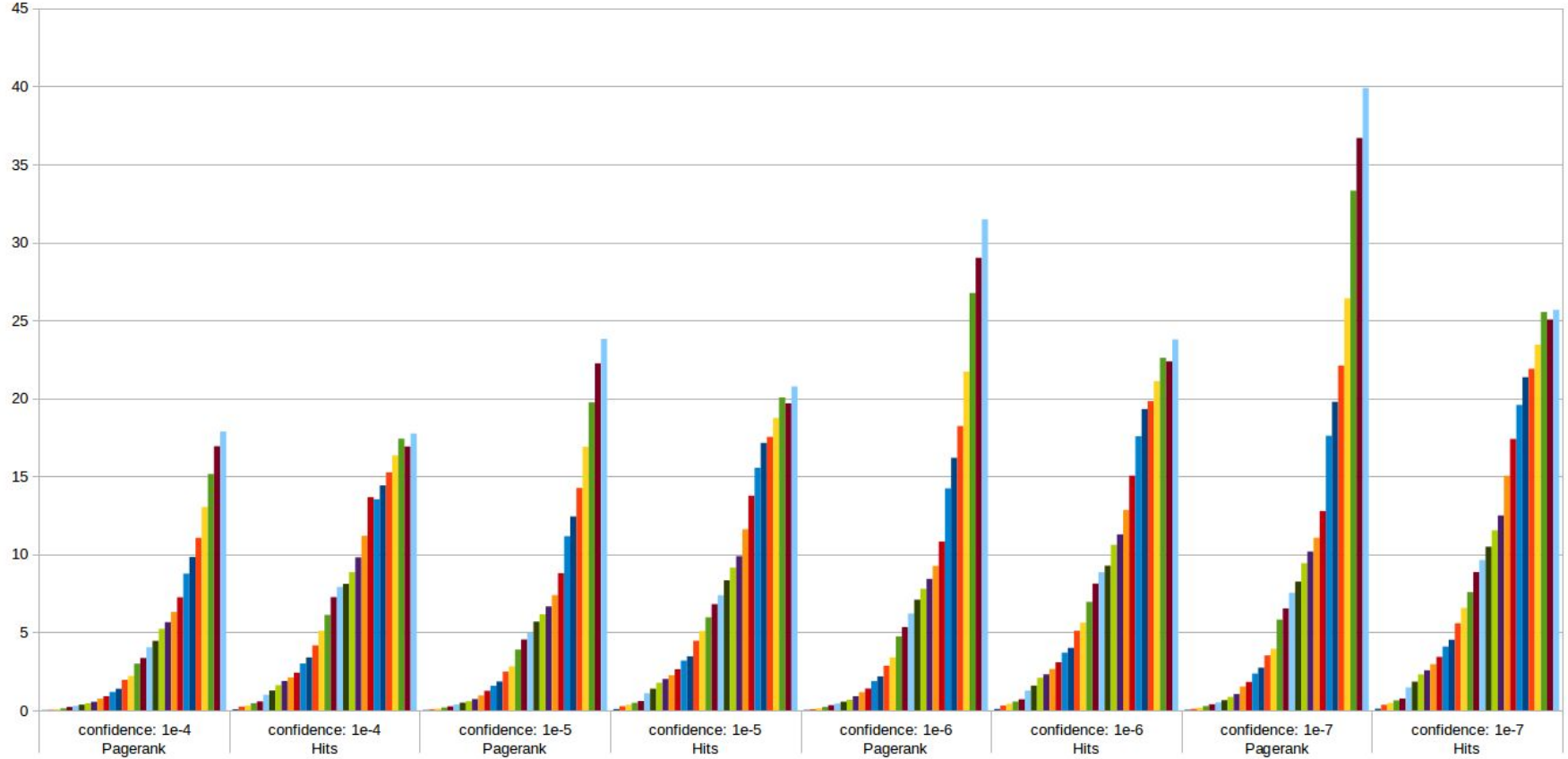
PageRank vs HITS

HITS weighted



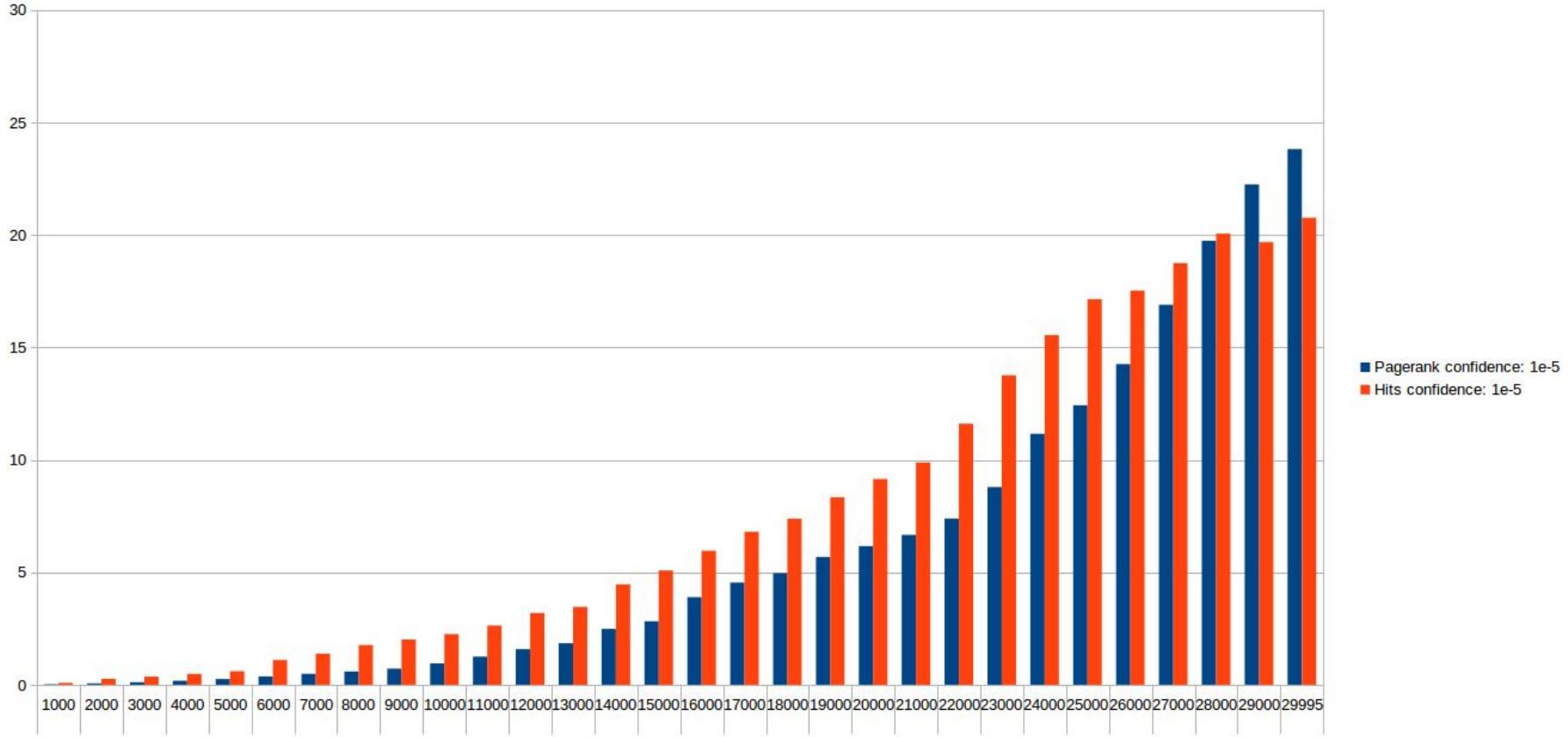
PageRank vs HITS

Average execution times on different graphs



PageRank vs HITS

Detailed average execution times



Matching

Best Match

Optimization of Best Match

Comparing the Results

Best Match

*“Given a query **q**, containing **n** query words, and a set of documents **S**, Best Match its a method that finds a subset of document S’ such that:*

- *each document s_i of S^i has a "reasonable" number of query words in it”*

Basic Steps:

- Counting how many query words the documents have
  - This value is called "score" of a document and it is at maximum *n*
- Ordering in decreasing order of score the documents (optional)
- Return all documents whose score is "reasonable"
  - set a threshold to define what is "reasonable"

Best Match

Two are the basic refinements to have a more efficient Best Match:

1. Using an inverted index

- in the form (word -> list of documents containing the word)
- the keys of the dataset are the query words
- we can have in O(1) all the documents with a determined word

2. Using the frequency instead of assigning score 1 to each query found

- defined as number of occurrences in document, n-term-occ(d)/n-word-doc(d)
- requires precalculation of occurrences for all words and all S
- it represents the relevance of documents to a particular word or query

Best Match

Computations step for the basic version of Best Match (BM):

1. The score of a document must depend on the frequency of a query term in that document (where the frequency is the ratio between the number of occurrences of the term and total number of words in the document).
2. Returns only the 20 documents with higher score.

Optimized Best Match

Computations step for the optimized version of Best Match (BM-OPT):

1. We have sorted query terms in decreasing order based on the cardinality of the set of documents where the term is contained;
2. While less than K = 0,2% of the documents have been scored, evaluate the score (based on the frequency) of all documents in the index of next term;
3. For the remaining query terms, update the score only for the K documents already scored in the previous phase;
4. Returns the 20 documents with higher score.

Results

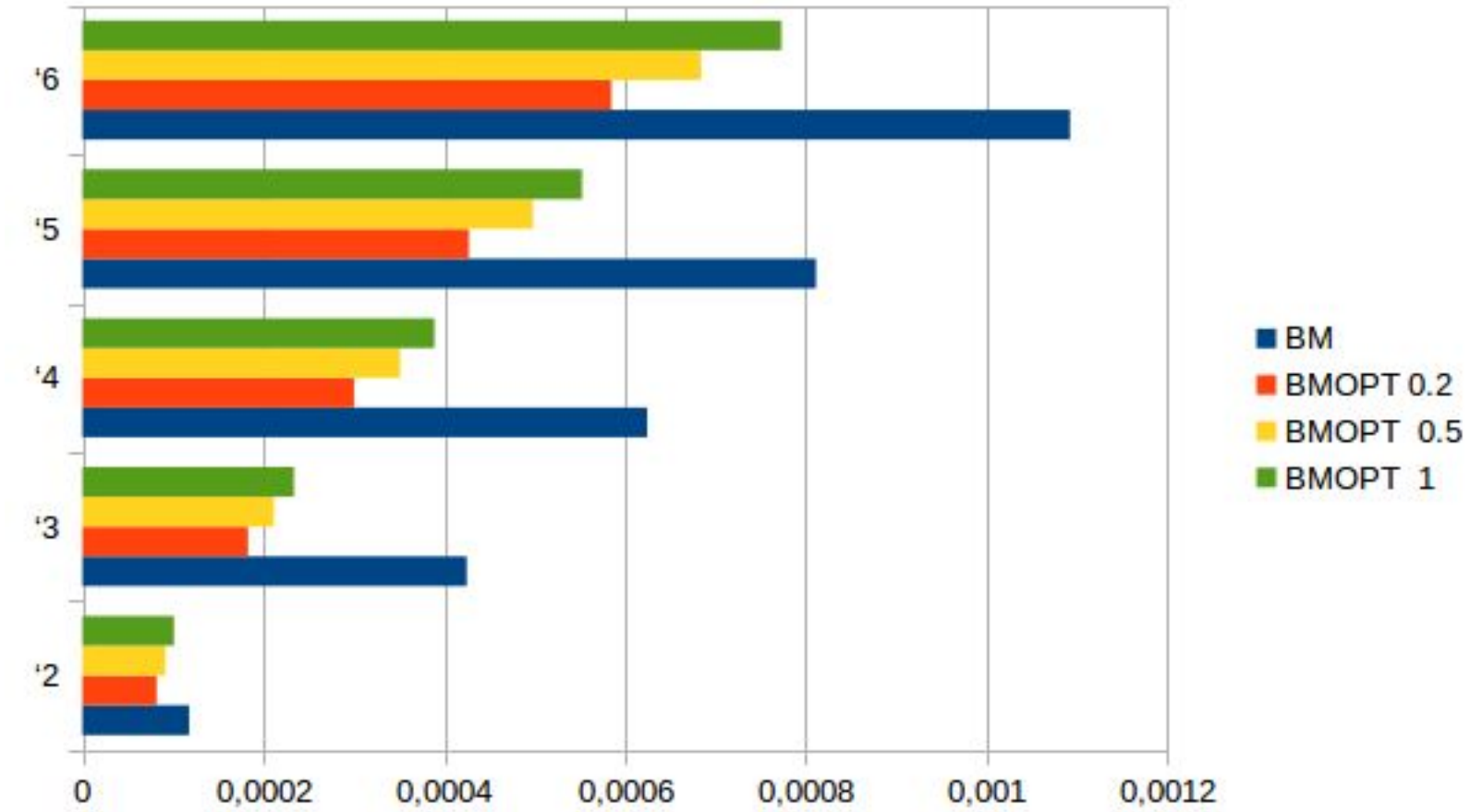
Experiment configuration

For obtaining the time comparison between **BestMatch** and its **Optimized Version** we run both algorithms on 1000 queries for each lenght in the range [2, 6], 10 times for each query:

- These 5000 queries has been randomly generated by the words of our Dataset.
- We also compare the time difference with different opt version k values
    - 0.2, 0.5, 1

Then we mean the results and plotted them on a graph.

Results


‘6
‘5
‘4
‘3
‘2
0
0,0002
0,0004
0,0006
0,0008
0,001
0,0012
BM
BMOPT 0.2
BMOPT 0.5
BMOPT 1

Search Engine

Comparing the Results

Search Engine

The implemented Search Engine combine the algorithms seen before.

Given a query:

1. Find the documents that match the query using:
    a. Best Match
    b. Optimized Best Match
2. Order these documents with score given by:
    a. Page Rank
    b. HITS
        i. authority, hubbiness, weighted

Each test it was repeated for all the ranking confidences ($1e^{-4}$,$1e^{-5}$,$1e^{-6}$,$1e^{-7}$)

Results

Experiment configuration

For fully analising the output of search engine we run it with 10 queries tuned on specific pages (K=0.2%):

Document	Score BM	Score BM op	BM	BMOPT	PageRank BM	Hits BM w	Hits BM a	Hits BM h	PageRank B	Hits BMOPT	Hits BMOPT	Hits BMOPT h
http://www.awn.com/news/e3-game-conference-remain-la	0.25	0.25	Y	Y	20	18	20	18	20	19	20	19
http://www.awn.com/news/sony-unveils-e3-lineup	0.2	0.2	Y	Y	12	13	13	12	13	16	15	15
http://www.ign.com/articles/2014/05/06/microsoft-sets-time-and-	0.1875	0.1875	Y	Y	7	11	10	11	6	14	11	14
http://www.ign.com/articles/2014/05/27/game-scoop-presents-th	0.173913043	0.173913043	Y	Y	5	9	4	10	4	10	4	13
http://www.awn.com/tag/microsoft	0.166666666	//////////////	Y	N	14	15	15	14				
http://www.awn.com/tag/e3	0.166666666	0.166666666	Y	Y	13	14	14	13	14	17	16	16
http://www.ign.com/events/e3?schedule	0.144508670	0.144508670	Y	Y	8	2	11	2	7	2	12	2
http://www.ign.com/events/e3	0.144508670	0.144508670	Y	Y	2	1	2	1	2	1	2	1
http://www.awn.com/news/star-trek-video-game-villain-revealed	0.142857142	0.142857142	Y	Y	18	17	18	17	18	18	18	18
http://www.awn.com/news/video-games-live-announces-e3-line	0.142857142	0.142857142	Y	Y	15	20	16	20	15	20	17	20
http://www.awn.com/news/ridley-scott-produce-halo-digital-featu	0.142857142	//////////////	Y	N	11	19	12	19				
http://www.awn.com/blog/how-get-god-s-eye-view-your-story-mi	0.142857142	//////////////	Y	N	16	16	17	15				
http://www.ign.com/wikis/e3/Games_at_E3_2014	0.135820895	0.135820895	Y	Y	3	8	3	9	3	9	3	11
http://www.ign.com/articles/2014/06/09/e3-2014-phantom-dust-r	0.133333333	0.133333333	Y	Y	9	4	5	5	8	4	6	5
http://www.tomshardware.com/articles/?brand=xbox	0.122807017	//////////////	Y	N	4	7	9	6				
http://www.ign.com/articles/2014/06/09/e3-2014-happy-wars-an	0.12	0.12	Y	Y	10	3	6	3	10	3	8	3
http://www.ign.com/videos?page=2&filter=games-trailer	0.116822429	0.116822429	Y	Y	19	10	19	8	19	12	19	10
http://www.ign.com/articles/2014/05/13/microsoft-reveals-xbox-o	0.111111111	//////////////	Y	N	6	5	7	4				
http://www.awn.com/news/video-games-live-announces-e3-mee	0.111111111	0.111111111	Y	Y	17	12	8	16	16	15	10	17
http://www.ign.com/xbox-one	0.111111111	0.111111111	Y	Y	1	6	1	7	1	8	1	8
http://www.ign.com/articles/2014/06/09/e3-2014-a-ton-of-new-i	//////////////	0.107142857	N	Y					11	7	9	6
http://www.ign.com/articles/2014/06/10/e3-2014-what-did-you-th	//////////////	0.105263157	N	Y					5	6	5	7
http://www.ign.com/articles/2014/06/10/e3-2014-xbox-boss-give	//////////////	0.105263157	N	Y					9	5	7	4
http://www.ign.com/wikis/e3/Test	//////////////	0.104895104	N	Y					17	13	13	12
http://www.ign.com/videos?page=2&filter=all	//////////////	0.104602510	N	Y					12	11	14	9

Results

K is an approximation

Let $w_1 \dots w_i \dots w_n$ the query words

I_i = documents set where w_i is in

$$\begin{cases} \left| \bigcap_i^n I_{wi} \right| < k \Rightarrow (BM = BMOPT) \\ \left| \bigcap_i^n I_{wi} \right| >= k \Rightarrow (BM != BMOPT \text{ v } BM > BMOPT) \end{cases}$$

Results

Experiment configuration

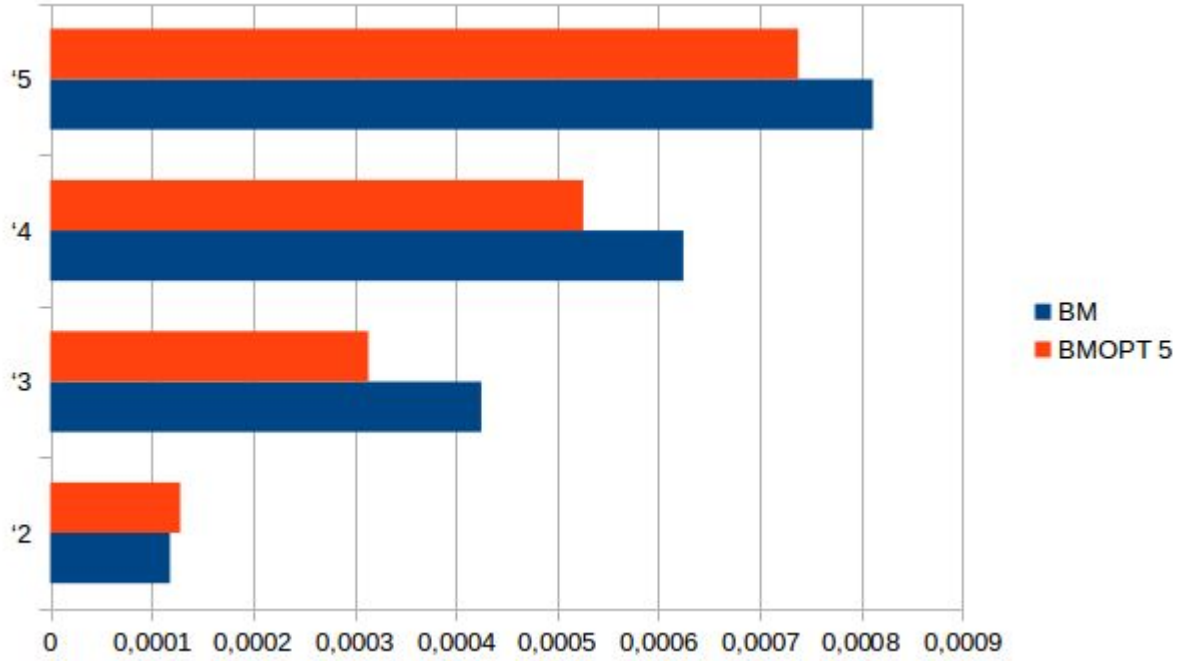
For fully analising the output of search engine we run it with 10 queries tuned on specific pages (K=5%):

Document	Score	BM	BMOPT
http://www.awn.com/news/e3-game-conference-remain-la	0.25	Y	Y
http://www.awn.com/news/sony-unveils-e3-lineup	0.2	Y	Y
http://www.ign.com/articles/2014/05/06/microsoft-sets-time-and-date-for-xbox-e3-2014-media-briefing	0.1875	Y	Y
http://www.ign.com/articles/2014/05/27/game-scoop-presents-the-xbox-e3-preview	0.173913043	Y	Y
http://www.awn.com/tag/microsoft	0.166666666	Y	Y
http://www.awn.com/tag/e3	0.166666666	Y	Y
http://www.ign.com/events/e3?schedule	0.144508670	Y	Y
http://www.ign.com/events/e3	0.144508670	Y	Y
http://www.awn.com/news/star-trek-video-game-villain-revealed-e3-conference	0.142857142	Y	Y
http://www.awn.com/news/video-games-live-announces-e3-lineup	0.142857142	Y	Y
http://www.awn.com/news/ridley-scott-produce-halo-digital-feature-xbox	0.142857142	Y	Y
http://www.awn.com/blog/how-get-god-s-eye-view-your-story-microsoft-excel	0.142857142	Y	Y
http://www.ign.com/wikis/e3/Games_at_E3_2014	0.135820895	Y	Y
http://www.ign.com/articles/2014/06/09/e3-2014-phantom-dust-returning-as-xbox-one-exclusive	0.133333333	Y	Y
http://www.tomshardware.com/articles/?brand=xbox	0.122807017	Y	Y
http://www.ign.com/articles/2014/06/09/e3-2014-happy-wars-announced-for-xbox-one	0.12	Y	Y
http://www.ign.com/videos?page=2&filter=games-trailer	0.116822429	Y	Y
http://www.ign.com/articles/2014/05/13/microsoft-reveals-xbox-one-without-kinect	0.111111111	Y	Y
http://www.awn.com/news/video-games-live-announces-e3-meet-greet-lineup	0.111111111	Y	Y
http://www.ign.com/xbox-one	0.111111111	Y	Y

Results

Experiment configuration

For fully analising the output of search engine we run it with 10 queries tuned on specific pages (K=5%):



Results

We show the difference between different confidences.
To clarify we showed the 20th page returned from the query seen before.

Confidences	Type	PageRank	Hits w	Hits a	Hits h
1E-4	BM	1	3	1	7
	BMOPT	1	3	1	8
1E-5	BM	1	6	1	7
	BMOPT	1	8	1	8
1E-6	BM	1	6	1	7
	BMOPT	1	8	1	8
1E-7	BM	1	6	1	7
	BMOPT	1	8	1	8

The modified ranks are in page with lowest score:

- 17th
- 18th
- 19th
- 20th
- 21th
- 22th
- 23th
- 24th
- 25th

The variation in all these pages is similar to the 20th page

Auction

First Price Auction

Generalized Second Price Auction

Results

First Price sealed-bid Auction

In this kind of auction, bidders submit simultaneous "**sealed bids**" to the seller. The terminology comes from the original format for such auctions, in which bids were written down and provided in sealed envelopes to the seller, who would then open them all together.

The highest bidder wins the object and pays the value of her bid.

First Price Auction

Non - truthfulness

In this type of auction, bidding your true value is not a dominant strategy. By bidding your true value, you would get a payoff of 0 if you lose, and you would also get a payoff of 0 if you win ($v_i=b_i$), since you'd pay exactly what it was worth to you.

The optimal way to bid in a first-price auction is to "**shade**" your bid slightly downward, so that if you win you will get a positive payoff.

- Determining how much to shade your bid involves balancing a trade-off between two opposing forces.

Generalized Second Price Auction

The sealed-bid second-price auction is particularly interesting, and there are a number of examples of it in widespread use. The auction form used on eBay is essentially a second-price auction. The pricing mechanism that search engines use to sell keyword-based advertising is a generalization of the second-price auction.

Bidders submit simultaneous sealed bids to the sellers.

The highest bidder wins the object and pays the value of the second-highest bid.

Generalized Second Price Auction

Truthfulness

One of the most important results in auction theory is the fact that with independent, private values, bidding your true value is a dominant strategy in a second price sealed-bid auction. That is, the best choice of bid is exactly what the object is worth to you.

The key point is that the value of *i*'s bid only affects whether *i* wins or loses, but never affects how much *i* pays in the event that she wins — the amount paid is determined entirely by the other bids, and in particular by the largest among the other bids.

Results

Bot description

We used the following bots:

- **Best-response** with balanced tie-breaking rule
- **Best-response-competitor-busting:** submits the highest possible bid that gives the desired slot.
- **Best-response-altruistic:** submits the lowest possible bid that gives the desired slot.
- **Competitor-bursting:** submits a bid greater than the highest bid seen in previous auction, even if it is greater than own value

Results

Bot description

- **Budget-saving** submits that is the minimum among the last non-winning bid and the advertiser value for the query
- **Random bot** submits a random value in a range [1, current budget] if current budget > initial-budget/2 otherwise in a range[1, current budget/2].

Our Bot:

- **Altruistic_budget** if I have more than half of the budget i use Best-response-altruistic else Budget-saving

Results

Bot description

- **Competitor_budget** if I have more than half of the budget i use competitor-bursting else Best-response-competitor
- **Threshold_budget** if adv_value is greater than a threshold and i have more than half of the budget i use Best-response-competitive else best-response. If lower than a threshold i use budget-saving
- **Preferential_budget** if I have more than half of the budget i use Best-response-competitive else best-response

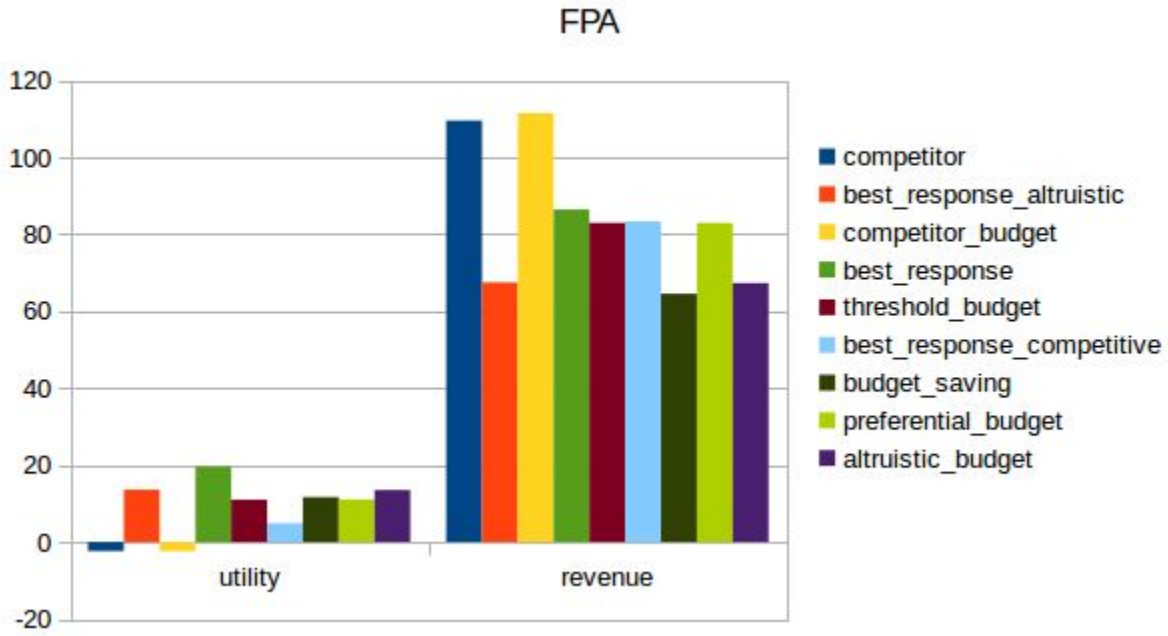
Results

Fixed Test configurations

- **Number of Query Words:** 1
- **Number of slots:** 2
  - **Slots #1 click-through:** 1
  - **Slots #2 click-through:** 0.8
- **Number of Auction:** 10
- **Number of Advertiser:** 3 (1 bot to test and 2 enemies of the same kind)
- **Values:** 10
- **Budgets:** 50
- **Number of runs:** 500

Results

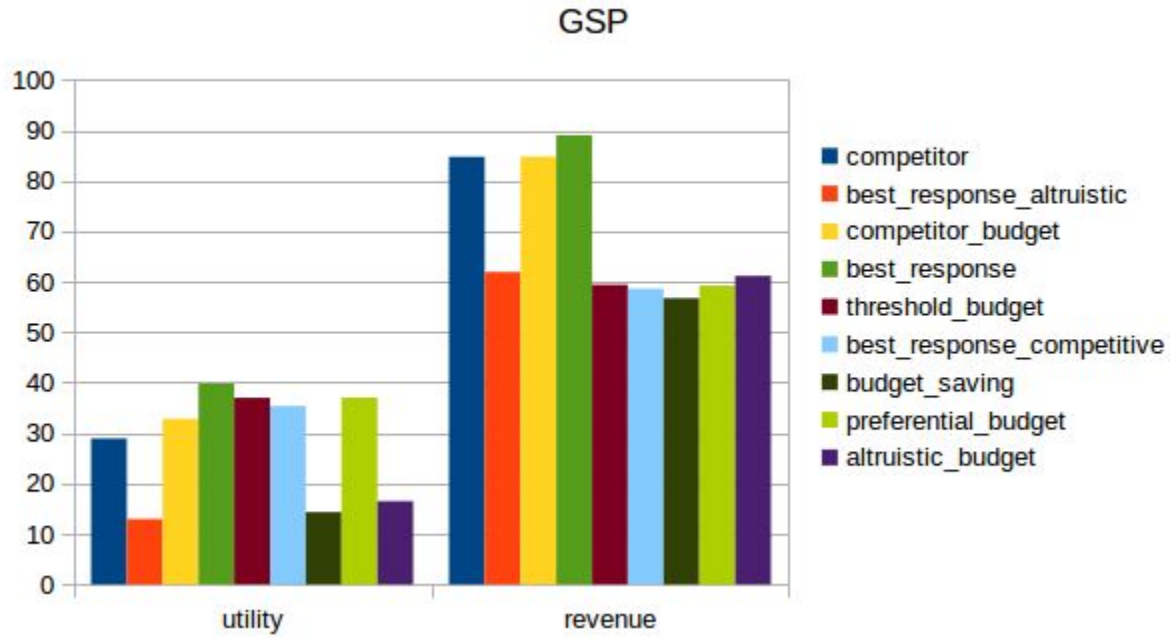
FPA



- **Advertiser-side (utility)**
- Due to the fact that FPA is not a truthful auction, and bots "savers" are rewarded by their behavior
- **Seller-side (revenue)**
- As we expected the bot "savers" give lower revenue than bots "aggressive"
- The behavior of competitor_budget is justified because it is actually a bot that acts like a competitor bots up to certain level of budget

Results

GSP


GSP
100
90
80
70
60
50
40
30
20
10
0
utility
revenue
competitor
best_response_altruistic
competitor_budget
best_response
threshold_budget
best_response_competitive
budget_saving
preferential_budget
altruistic_budget

Results

GSP considerations

- **Advertiser-side**
  - In GSP all bots have a higher utility than FPA
  - We notice that the bots that offer bids near to their “real” value perform better, obtaining a higher utility.
  - The explanation for these results is that the GSP is a truthful auction
  - The bots that tend to lower the value of its bid in the end earn less utility
- **Seller-side**
  - As well FPA in GSP bots that generates more revenue is the competitor, best_response and competitor_budget

Results

Most interesting bot

Observing statics tests we have seen that:

- **Advertiser-side**
  - Best_response bot gains higher utility in both auctions.
- **Seller-side**
  - Competitor e best_response_competitor generate high revenues in both auctions

to confirm what we have seen previously we have performed some tests with random values

Results

Test random configuration

- **Number of Query Words:** 3
- **Number of slots:** [2,3]
- **Slots click-through:** [0,1]
- **Number of Auction:** 100
- **Number of Advertiser:** 3 (1 bot to test and 2 enemies of the same kind)
- **Values:** [0,100]
- **Budgets:** [50,200]
- **Number of runs:** 500

Results

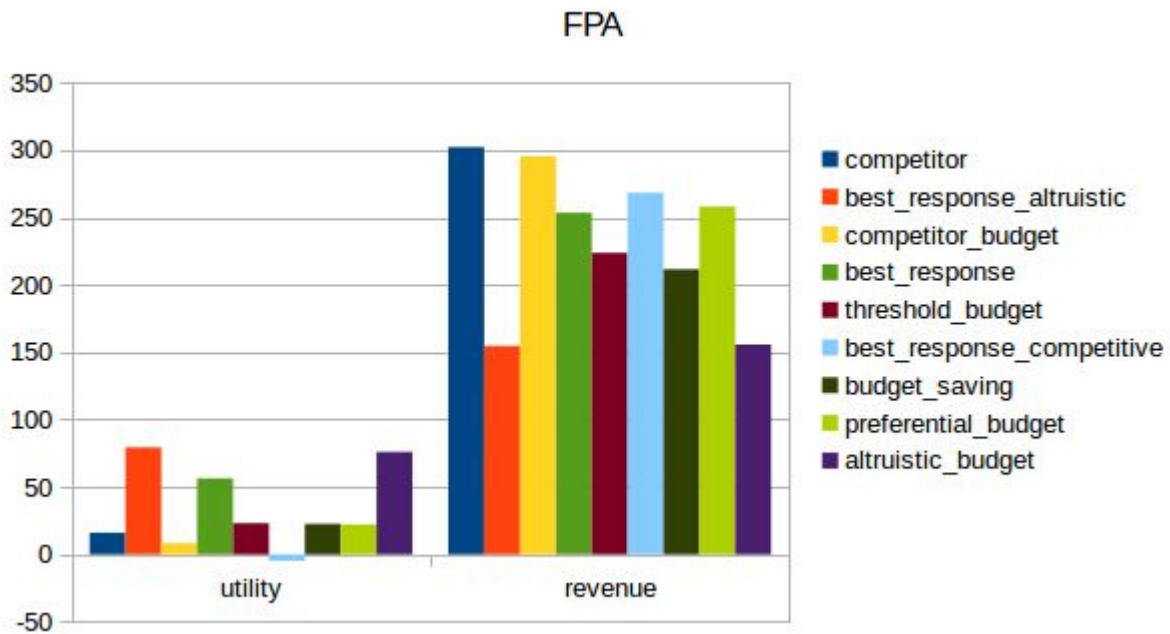
Test random configuration

In these tests the random bot covers a fundamental role

This bot significantly can reduce or raise the opponent's average utility because he makes random choices with not much sense

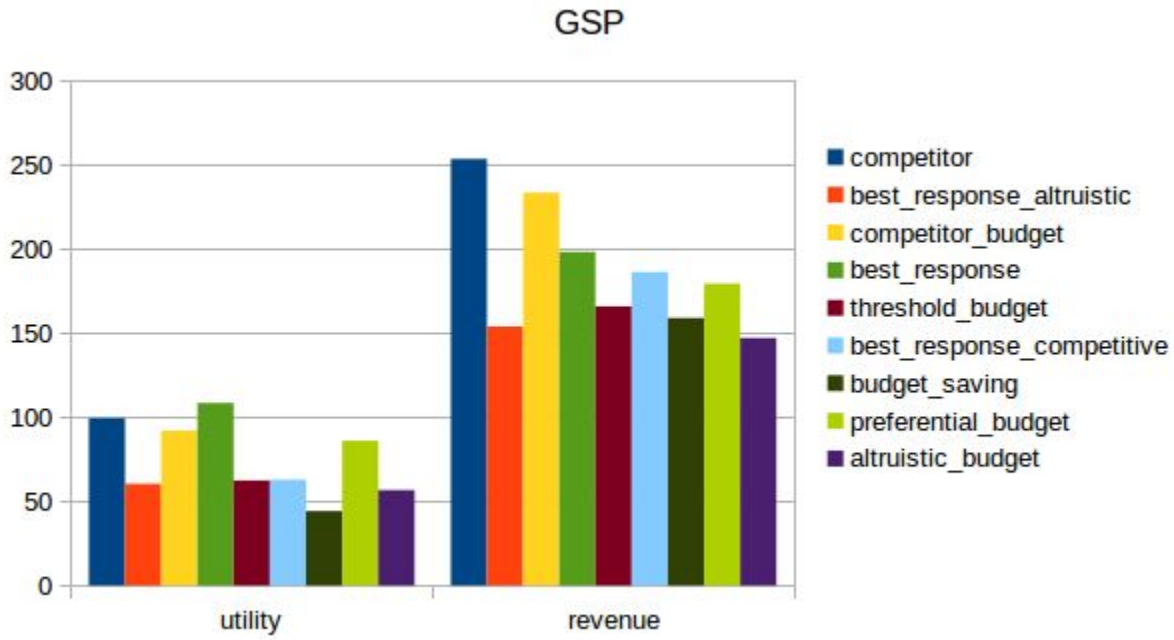
Results

FPA


FPA
350
300
250
200
150
100
50
0
-50
utility
revenue
competitor
best_response_altruistic
competitor_budget
best_response
threshold_budget
best_response_competitive
budget_saving
preferential_budget
altruistic_budget

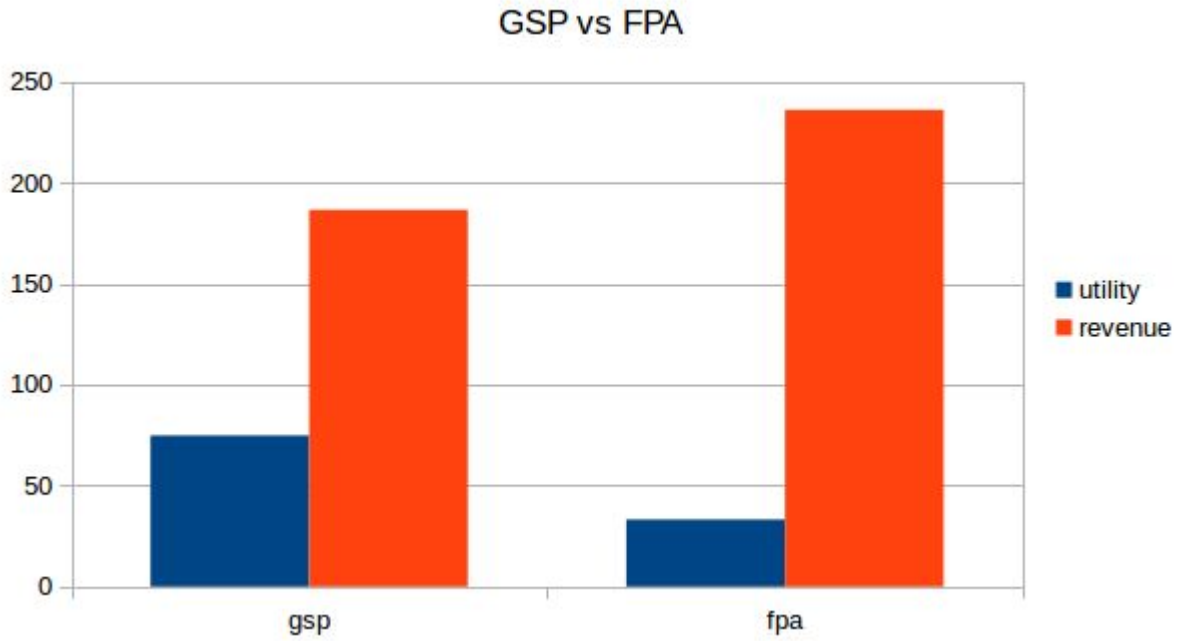
Results

GSP


GSP
300
250
200
150
100
50
0
utility
revenue
competitor
best_response_altruistic
competitor_budget
best_response
threshold_budget
best_response_competitive
budget_saving
preferential_budget
altruistic_budget

Results

GSP vs FPA



Results

GSP vs FPA

we can observe that:

- FPA lead to a higher revenue compared to GSP
- GSP hold a higher utility and heuristic in relation to FPA

Conclusion

Summing up

Conclusion

Observing our experiments, a good configuration to implement a search engine is:

- **Ranking**: HITS, gives more informations about a node and fastest execution times with a huge graphs
- **Matching**: Best Match, for its precision (BM-OPT for lower running time)
- **Auction**:
  - For the advertiser GSP
  - For the seller FPA